UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/784,280	02/24/2004	Kumanan Yogaratnam	50605-1 /aba	9969

7380 7590 01/10/2008

SMART & BIGGAR
P.O. BOX 2999, STATION D
900-55 METCALFE STREET
OTTAWA, ON K1P5Y6
CANADA

EXAMINER	
WANG, JUE S	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
01/10/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/784,280	YOGARATNAM ET AL.	
	Examiner	Art Unit	
	Jue S. Wang	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 February 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *08/11/2005*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-42 have been examined.

Claim Objections

2. Claim 9 is objected to because of the following informalities:

3. Claim 9, lines 5-7, the phrase "accessible to the internal software module any other external software modules" should read "accessible to the internal software module and any other external software modules".

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 1-42 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

A. The following claim language is not clear and indefinite:

i. As per claims 1, 23, 36, 41, and 42, the terms "software element" and "software module" are used. It is not clearly understood how software elements different from software module since a software module can certainly be considered as a software element.

ii. As per claims 1, 23, 36, 41, and 42, the terms "internal" and "external" are used. These terms are not clearly understood because it is not clear what is

considered as "internal" and "external" (i.e., does internal mean the software elements are found inside a specific application, inside a specific memory region or storage device, and does external mean that the software elements are found outside a specific application, outside a specific memory region or storage device, or does internal refer to elements that are originally packaged with a specific application, whereas external refer to elements are later imported into the application?).

iii. As per claim 16, lines 1-2, the term "the at least one software module" is used. This limitation is not clearly understood because it is not clear if the software module is an internal software module or an external software module.

iv. As per claim 19, it is uncertain whether this is intended to be independent of dependent claim. Claim 19 is a device claim, but it depends on claim 1 which is a method claim.

v. As per claims 20 and 31, it is uncertain whether these are intended to be independent of dependent claims. Claims 20 and 31 are computer readable medium claims, but they depend on claims 1 and 23 which are method claims.

vi. As per claim 32, the terms "software element" and "software code" are used. It is not clearly understood how software elements different from software code since software code can certainly be considered as a software element.

vii. As per claim 41, lines 32-33, claim 42, lines 25-26, the phrase "providing access to the electronic content by the external software module" is used. This limitation is not clearly understood because it is not clear how this phrase should

be interpreted (i.e., is the electronic content being provided to the external software module so that the external software module can access the electronic content, or is the external software module provide by the external software module so that other software modules can access the electronic content?).

Appropriate corrections are required.

Any claim not specifically addressed, above, is being rejected as incorporating the deficiencies of a claim upon which it depends.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7. Claims 32-40 and 42 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

8. Claim 32 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. In claim 32, a "system" is recited; however, it appears that the system would reasonably be interpreted by one of ordinary skill in the art as software, per se, since the software element managers and electronic content loader recited as part of the system would reasonably be interpreted by one of ordinary skill in the art as software, per se. As such, it is believed that the system of claim 32 is reasonably interpreted as functional descriptive

material, per se, failing to be tangibly embodied or include any recited hardware as part of the system.

9. Claims 33-35 fail to resolve the deficiencies of claim 32. Claims 33-35 disclose additional features of the software element managers. The limitations recited in claims 33-35 do not invalidate the reasonable interpretation of the system as software, per se, by one of ordinary skill in the art.

10. Claim 36 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. In claim 36, a "system" is recited; however, it appears that the system would reasonably be interpreted by one of ordinary skill in the art as software, per se, since the internal software module and external software module recited as part of the system would reasonably be interpreted by one of ordinary skill in the art as software, per se. As such, it is believed that the system of claim 36 is reasonably interpreted as functional descriptive material, per se, failing to be tangibly embodied or include any recited hardware as part of the system.

11. Claims 37-40 fail to resolve the deficiencies of claim 32. Claims 37-40 disclose additional features of the internal software module and external software module. The limitations recited in claims 37-40 do not invalidate the reasonable interpretation of the system as software, per se, by one of ordinary skill in the art.

12. Claim 42 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. In claim 40, a "system" is recited; however, it appears that the system would reasonably be interpreted by one of ordinary skill in the art as software, per se, since the means for providing, means for processing, and means for detecting recited as part of the system would reasonably be interpreted by one of ordinary skill in the art as software, per se. As such, it is believed that the system of claim 42 is reasonably interpreted as functional descriptive material, per se, failing to be tangibly embodied or include any recited hardware as part of the system.

Claim Rejections - 35 USC § 102

13. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

14. Claims 23, 24, 31, 36, and 37 are rejected under 35 U.S.C. 102(e) as being anticipated by Jennings (US 6,717,593 B1).

15. As per claim 23, Jennings teaches the invention as claimed, including a method of processing electronic content comprising:

providing an internal software module defining embeddable software elements (see Fig 3, items 130, 310, Fig 7, step 404, column 2, lines 13-21, column 3, lines 2-12, column 6, lines 7-

10, column 8, lines 26-65, column 9, line 61 – column 10, line 51, column 11, lines 15-26; EN: the interactor is the internal software module and elements contained in the starting user interface document are considered internal software element because they are defined in the configuration file for the interactor);

providing an external software module (see column 2, lines 30-40, column 7, lines 2-11, column 9, line 54 – column 11, line 26; EN: layout and connector plug-ins downloaded after the initial download based on URLs in the configuration file interactor are considered as external software modules);

processing the electronic content using the internal software module to resolve embedded references in the electronic content to the embeddable software elements (see Figs 7-9, column 9, line 54 – column 11, line 26; EN: the parse tree of nodes is considered as the electronic content and the name of the nodes in the parse tree are embedded references);

notifying the external software module of the processing of the electronic content (column 11, lines 3-9; EN: the layout objects are notified of the processing when it is invoked to process the node); and

accessing the electronic content by the external software module responsive to the notifying (see column 11, lines 3-9).

16. As per claim 24, Jennings teaches manipulating the electronic content by the external software module (see column 11, lines 3-9).

17. As per claim 31, Jennings teaches a computer readable medium storing statements or instructions which when executed by a processor perform the method of claim 23 (see Fig 2, item 201, column 4, lines 42-52).

18. As per claim 36, Jennings teaches the invention as claimed, including a system comprising:

an internal software module defining embeddable software elements and configured to process electronic content by resolving references in the electronic content to the embeddable software elements (see Fig 3, items 130, 310, Figs 7-9, column 2, lines 13-21, column 3, lines 2-12, column 6, lines 7-10, column 8, lines 26-65, column 9, line 54 – column 11, line 26); and

an external software module configured to detect the processing of the electronic content and to access the electronic content responsive to the detection (see column 2, lines 30-40, column 7, lines 2-11, column 8, lines 40-45, column 9, line 54 – column 11, line 26).

19. As per claim 37, Jennings teaches the internal software module is further configured to supply to the external software module a notification that the electronic content is being processed and a handle associated with the electronic content, and wherein the external software module is configured to detect the processing by receiving the notification and to access the electronic content using the handle (see column 11, lines 3-9).

Claim Rejections - 35 USC § 103

20. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

21. Claims 1-4, 6-11, 15-17, 20-22, 25, 32-35, and 38-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jennings (US 6,717,593 B1), in view of Peterka et al., (WO 00/30346, hereinafter Peterka).

22. As per claim 1, Jennings teaches the invention as claimed, including a method of processing electronic content (see abstract) comprising:

providing an internal software module defining at least one internal software element (see Fig 3, items 130, 310, Fig 7, step 404, column 2, lines 13-21, column 3, lines 2-12, column 6, lines 7-10, column 8, lines 26-65, column 9, line 61 – column 10, line 51, column 11, lines 15-26; EN: the interactor is the internal software module and elements contained in the starting user interface document are considered internal software element because they are defined in the configuration file for the interactor);

providing registry for different types of software elements, the registry comprising a software element registry for external software elements (see Fig 3, item 310, column 6, lines 45-47, column 9, lines 23-61, column 11, lines 15-26; EN: the object model is considered a registry,

and elements from user and feature description documents downloaded after the initial download based on URLs in the configuration file are considered as external software elements); and

processing electronic content using the internal software module to resolve embedded references in the electronic content to the internal software elements and embedded references in the electronic content to any external software elements in the software element registry (see Figs 7-9, column 9, line 54 – column 11, line 26; EN: the parse tree of nodes is considered as the electronic content and the name of the nodes in the parse tree are embedded references),

wherein the external software elements in the software element registry comprise at least one external software element supported by software code (see column 7, lines 2-11; EN: layout objects are the software code supporting the software elements), wherein processing to resolve embedded references in the electronic content to the at least one external software element comprises supplying a handle associated with the electronic content, and wherein the electronic content is accessible to the software code using the handle (see column 8, lines 40-45, column 11, lines 3-9; EN: the reference to the node and the node's subtree of nodes is considered a handle to the electronic content).

Jennings only teaches providing one registry, and does not teach providing a plurality of registries.

Peterka teaches an API for providing a plurality of registries for different types of resources (see page 5, lines 1-3, 17-23, page 12, line 21 – page 13, line 4).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings to use the API taught by Peterka to divide the existing single registry (i.e., object model) into a plurality of registries to provide better organization of the different

types of objects that are currently registered in the object model such as user interface definition elements, layout objects, and connector objects (see column 2, lines 35-40, column 6, lines 45-47, column 9, lines 23-61 of Jennings).

23. As per claim 2, Jennings teaches providing a plurality of external software elements (see Figs 2, 3, column 4, lines 53-63); and registering each of the plurality of external software elements in the software element registry (see column 6, lines 45-47, column 8, lines 53-61).

24. As per claim 3, Jennings teaches the at least one internal software element comprises an internal visual element having a visual component for display, and wherein the at least one external software element comprises an external visual element having a visual component for display (see column 5, lines 52-65).

25. As per claim 4, Jennings teaches that each external software element is supported by software code (see column 7, lines 2-8), and wherein registering comprises providing registration information for each external software element from its supporting software code to the software element registry (see abstract, lines 16-18, column 2, lines 30-40).

26. As per claim 6, Jennings teaches the method further comprises:

requesting each embedded external software element from the software element registry (see column 11, lines 15-22);

determining whether the embedded external software element has been registered in the software element registry (See column 11, lines 15-22); and

returning a representation of the external software element where the embedded external software element is registered in the software element registry (see column 11, lines 24-27).

27. As per claim 7, Jennings teaches the software code manipulates the electronic content using the handle (see column 11, lines 3-9).

28. As per claim 8, Jennings teaches the plurality of registries further comprises a software module registry for software modules (see column 2, lines 35-40, column 9, lines 23-61).

29. As per claim 9, Jennings teaches providing at least one external software module (see column 2, lines 30-40, column 7, lines 2-11); and

registering with the software module registry each external software module that is to be accessible to the internal software module and any other external software modules (see column 2, lines 30-40, column 6, line 60 – column 7, line 11, column 9, lines 42-47).

30. As per claim 10, Jennings does not teach registering the internal software module with the software module registry.

Peterka teaches registering resources including software modules and plug-in modules in registries (see page 5, line 16 – page 6, line 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings to register the internal software module with the software module registry as suggested by Peterka because the registry is a convenient place where an application can learn about the resources that are available (see page 5, lines 16-26 of Peterka) and applications rely on the interactor to provide an user interface (see Fig 3 of Jennings), so it would be advantageous to register the interactor in a registry so that applications can learn the availability of the interactor.

31. As per claim 11, Jennings teaches wherein the at least one external software module comprises an external software module supporting an external software element (see column 7, lines 2-11), further comprising:

registering the external software element with the software element registry (see column 6, lines 45-47, column 9, lines 23-61, column 11, lines 15-26).

32. As per claim 15, Jennings teaches wherein the plurality of registries further comprises a script function registry for script functions (see column 6, lines 35-40, column 8, lines 53-58, column 10, lines 4-6).

33. As per claim 16, Jennings teaches wherein the at least one software module comprises a software module that defines at least one script function (see Fig 4, column 10, lines 4-6), further comprising:

registering with the script function registry each of the at least one script function that is to be accessible to the electronic content (see column 6, lines 35-40, column 8, lines 53-58).

34. As per claim 17, Jennings teaches that the electronic content comprises a UI (user interface) definition for a software application (see column 2, lines 1-22).

35. As per claim 20, Jennings teaches a computer readable medium storing statements or instructions which when executed by a processor performed the method of claim 1 (see Fig 2, item 201, column 4, lines 42-52).

36. As per claim 21, Jennings teaches calling a script function referenced in electronic content using a content handle (see column 9, lines 18-20, column 10, lines 4-6), wherein the electronic content is accessible using the handle (see column 9, lines 18-20, column 10, lines 4-6, column 11, lines 37-40).

37. As per claim 22, Jennings does not teach that the software module registry supports a query function for querying availability of software modules, and wherein registering with the script function registry comprises registering the query function with the script function registry.

Peterka teaches an API for providing a plurality of registries, where the registry supports a query function for querying availability of software modules (see page 5, lines 12-15, page 15, lines 3-19), and wherein registering modules with the registry comprises registering the query

function with the registry (see page 5, lines 12-15, page 11, lines 11-26, page 20, line 24 – page 23, line 10).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings to use the API taught by Peterka to support a query function for querying the availability of software modules because the query function can be used to check whether the module is installed (see page 22, lines 19-23 of Peterka).

38. As per claim 25, Jennings teaches providing a software module registry (see column 2, lines 35-40, column 9, lines 23-61) and registering the external software module with the internal software module to receive a notification of the processing of the electronic content (see column 2, lines 30-40, column 6, line 60 – column 7, line 11, column 9, lines 42-47)

Jennings does not teach registering an interface of the internal software module with the software module registry.

Peterka teaches registering resources including software modules and plug-in modules in registries (see page 5, line 16 – page 6, line 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings to register the internal software module with the software module registry as suggested by Peterka because the registry is a convenient place where an application can learn about the resources that are available (see page 5, lines 16-26 of Peterka) and applications rely on the interactor to provide an user interface (see Fig 3 of Jennings), so it would be advantageous to register the interactor in a registry so that applications can learn the availability of the interactor.

39. As per claim 32, Jennings teaches the invention as claimed, including a system comprising:

a software element manager maintaining a registry of software elements of different types including embeddable externally defined software elements (see Fig 3, item 310, column 6, lines 45-47, column 9, lines 23-61, column 11, lines 15-26; EN: the object model is considered a software element manager maintaining a registry, and elements from user and feature description documents downloaded after the initial download based on URLs in the configuration file are considered as embeddable external software elements); and

an electronic content loader defining embeddable internal software elements, configured to receive electronic content, to resolve references in the electronic content to the embeddable internal software elements, and to resolve references in the electronic content to any embeddable external software elements in the registry of the embeddable software element manager element (see Fig 3, items 130, 310, Figs 7-9, column 2, lines 13-21, column 3, lines 2-12, column 6, lines 7-10, column 8, lines 26-65, column 9, line 54 – column 11, line 26; EN: the interactor is the internal software module and elements contained in the starting user interface document are considered internal software element because they are defined in the configuration file for the interactor, the parse tree of nodes is considered as the electronic content and the name of the nodes in the parse tree are embedded references),

wherein the embeddable external software elements in the registry of the embeddable software element manager comprise at least one embeddable external software element supported by software code, wherein the electronic content loader is further configured to resolve the references in the electronic content to the at least one embeddable external software element

by supplying a handle associated with the electronic content, and wherein the electronic content is accessible to the software code using the handle (see column 8, lines 40-45, column 11, lines 3-9; EN: the reference to the node and the node's subtree of nodes is considered a handle to the electronic content).

Jennings only teaches providing one registry for maintaining software elements of different types, and do not teach a plurality of software element managers, each maintaining a registry of a software elements of a predetermined type.

Peterka teaches an API for providing a plurality of resource manager for managing resources of a predetermined type of resource (see page 15, lines 3-19) and a plurality of registries for different types of resources (see page 5, lines 1-3, 17-23, page 12, line 21 – page 13, line 4).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings to use the API taught by Peterka to divide the existing single manager and registry (i.e., object model) into a plurality of managers and registries to provide better organization of the different types of objects that are currently registered in the object model such as user interface definition elements, layout objects, and connector objects (see column 2, lines 35-40, column 6, lines 45-47, column 9, lines 23-61 of Jennings).

40. As per claim 33, Jennings teaches a plurality of external software modules comprising an external software module defining at least one of the embeddable externally defined software elements (see column 7, lines 2-8),

wherein the plurality of software element managers further comprises a software module manager maintaining a registry of any of the external software modules that are to be accessible to the content loader and other external software modules (see column 2, lines 30-40, column 6, line 60 – column 7, line 11, column 9, lines 42-47).

41. As per claim 34, Jennings teaches the plurality of software element managers further comprises a script function manager maintaining a registry of script functions (see column 6, lines 35-40, column 8, lines 53-58, column 10, lines 4-6), supported by at least one of the content loader and any of the external software modules, that are to be accessible to the electronic content (see column 6, lines 35-42, 57-60, column 9, lines 18-20).

42. As per claim 35, Jennings teaches wherein the plurality of software element managers provides a defined common interface for the electronic content (see Fig 3, column 6, lines 60-66), and wherein the processor is further configured to provide a custom interface between a native computer system and the plurality of software element managers (see Fig 3, column 6, lines 60-66).

43. As per claim 38, Jennings teaches the system further comprising:

a plurality of external software modules including the external software module (see column 2, lines 30-40, column 7, lines 2-11); and

a software module manager maintaining a registry of software modules (see column 2, lines 35-40, column 9, lines 23-61),

wherein the at least one of the external software modules is further configured to register with the software module manager (see column 2, lines 30-40, column 6, line 60 – column 7, line 11, column 9, lines 42-47).

Jennings does not teach registering the internal software module with the software module registry.

Peterka teaches registering resources including software modules and plug-in modules in registries (see page 5, line 16 – page 6, line 2).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings to register the internal software module with the software module registry as suggested by Peterka because the registry is a convenient place where an application can learn about the resources that are available (see page 5, lines 16-26 of Peterka) and applications rely on the interactor to provide an user interface (see Fig 3 of Jennings), so it would be advantageous to register the interactor in a registry so that applications can learn the availability of the interactor.

44. As per claim 39, Jennings teaches that the at least one of the external software module is configured to register with the software module manager upon detection of a registration trigger (see column 9, lines 23-25, 59-61). While Jennings does not explicitly teach that the internal software module is configured to register upon detection of a registration trigger, it would have been obvious to one of ordinary skill in the art that the registration trigger used for the external software modules can be applied to the internal software module when it is being initialized (see

column 9, lines 55-61) since the initialization of the internal software module signifies that the module is available to other applications.

45. As per claim 40, Jennings teaches that the registration trigger is a initialization trigger (see column 9, lines 23-25, 59-61).

46. As per claim 41, Jennings teaches the invention as claimed, including an electronic device comprising:

a processor (see Fig 2, item 202); and

a computer-readable medium accessible by the processor and storing instructions which when executed by the processor perform the method (see Fig 2, item 201, column 4, lines 42-52) comprising:

providing an internal software module defining an embeddable internal software element (see Fig 3, items 130, 310, Fig 7, step 404, column 2, lines 13-21, column 3, lines 2-12, column 6, lines 7-10, column 8, lines 26-65, column 9, line 61 – column 10, line 51, column 11, lines 15-26; EN: the interactor is the internal software module and elements contained in the starting user interface document are considered internal software element because they are defined in the configuration file for the interactor);

providing registry for different types of software elements, the registry comprising a software element registry for an externally defined embeddable external software element (see Fig 3, item 310, column 6, lines 45-47, column 9, lines 23-61, column 11, lines 15-26; EN: the object model is considered a registry, and elements from

user and feature description documents downloaded after the initial download based on URLs in the configuration file are considered as external software elements); and

processing electronic content using the internal software module to resolve any embedded references in the electronic content to the embeddable internal software element and any embedded references in the electronic content to the embeddable external software element in the embeddable software element registry (see Figs 7-9, column 9, line 54 – column 11, line 26; EN: the parse tree of nodes is considered as the electronic content and the name of the nodes in the parse tree are embedded references), wherein the embeddable external software element is supported by software code, wherein processing electronic content to resolve any embedded references in the electronic content to the embeddable external software element comprises supplying a handle associated with the electronic content, and wherein the electronic content is accessible to the software code using the handle (see column 8, lines 40-45, column 11, lines 3-9; EN: the reference to the node and the node's subtree of nodes is considered a handle to the electronic content);

providing an external software module (see column 2, lines 30-40, column 7, lines 2-11, column 9, line 54 – column 11, line 26; EN: layout and connector plug-ins downloaded after the initial download based on URLs in the configuration file interactor are considered as external software modules);

detecting the processing of the electronic content (column 11, lines 3-9; EN: the layout objects detect the processing when they are invoked to process the node); and

providing access to the electronic content by the external software module (see column 11, lines 3-17).

Jennings only teaches providing one registry, and does not teach providing a plurality of registries.

Peterka teaches an API for providing a plurality of registries for different types of resources (see page 5, lines 1-3, 17-23, page 12, line 21 – page 13, line 4).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings to use the API taught by Peterka to divide the existing single registry (i.e., object model) into a plurality of registries to provide better organization of the different types of objects that are currently registered in the object model such as user interface definition elements, layout objects, and connector objects (see column 2, lines 35-40, column 6, lines 45-47, column 9, lines 23-61 of Jennings).

47. As per claim 42, this is a system claim whose limitations are substantially similar to claim 41. Therefore, it is rejected using the same reasons as claim 41.

48. Claims 5, 12-14, and 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jennings (US 6,717,593 B1), in view of Peterka et al., (WO 00/30346, hereinafter Peterka), as applied to claims 2, 11, and 25 above, further in view of Fernando et al. (US 2004/0034860, hereinafter Fernando).

49. As per claim 5, Jennings teaches the registration information comprises an external software element name (see column 6, lines 7 – 13, column 10, lines 8-13, 41-50).

Jennings does not specifically teach the registration information includes a pointer to a portion of the software code supporting the external software element.

Fernando teaches a framework for dynamically extending applications using extension objects (see abstract), where the extension objects are registered in a registry and registry information includes a pointer to the extension object (see [0005], [0033]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings and Peterka to maintain a pointer to a portion of the software code as suggested by Fernando because the software code must be accessed from the registry to process the electronic content (see column 10, lines 62-65, column 11, lines 3-9 of Jennings) and a pointer is a well known technique in the art to provide a reference to a piece of code.

50. As per claim 12, Jennings teaches that the internal software module supports a notify function to notify the at least one external software module that electronic content is being processed (see column 11, lines 5-9).

Jennings and Peterka do not teach registering for the notify function each external software module that is to be notified that electronic content is being processed; providing a notification to each external software module registered for the notify function that electronic content is being processed.

Fernando teaches that the application manager has a notify function to provide notification to extension objects, including registering for the notification function each

extension object that is to be notified, and providing a notification to each extension object registered for the notify function (see [0053]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings and Peterka to register for the notify function each external software module that is to be notified that the electronic content is being processed and providing a notification to each external software module registered for the notify function that the content is being processed as taught by Fernando because notifications can be provided to a selected set of modules based on the type of the notification (see [0053] of Fernando).

51. As per claim 13, Jennings teaches providing a notification comprises supplying a content handle using which the electronic content is accessible (see column 8, lines 40-45, column 11, lines 5-9).

52. As per claim 14, Jennings teaches at least one of the external software module registered for the notify function accesses and manipulates the electronic content using the content handle (see column 11, lines 5-9).

53. As per claim 26, Jennings teaches providing an external software module comprises providing a plurality of external software modules (see column 2, lines 30-40, column 7, lines 2-11), and registering with the internal software module using the interface of each of the plurality of the external software modules that is to receive a notification (see column 2, lines 30-40, column 6, line 60 – column 7, line 11, column 9, lines 42-47).

Jennings teaches that the notification is provided by the internal module to the appropriate external software module, and Jennings does not teach notifying each external software module registered with the internal software module.

Fernando teaches that an application manager having a notify function to provide notification to extension objects, including registering for the notification function each extension object that is to be notified, and providing a notification to each extension object registered for the notify function (see [0053]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Jennings and Peterka to provide a notification to each external software module registered for the notify function that the content is being processed as taught by Fernando because notifications can be provided to a selected set of modules based on the type of the notification (see [0053] of Fernando), thus shifting the burden of selecting the appropriate external module to notify from the internal module to the external modules.

54. As per claim 27, Jennings teaches registering with the software module registry each of the plurality of external software modules that is to be accessible to the internal software module and other external software module (see column 2, lines 30-40, column 6, line 60 – column 7, line 11, column 9, lines 42-47).

55. As per claim 28, Jennings teaches wherein the plurality of external software modules comprises software modules supporting embeddable external software elements (see column 7, lines 2-11), further comprising:

providing a software element registry for the embeddable external software elements (see Fig 3, item 310, column 6, lines 45-47, column 9, lines 23-61, column 11, lines 15-26); and

registering with the software element registry each of the embeddable external software elements that is to be accessible to the electronic content (see column 6, lines 45-47, column 8, lines 53-61),

wherein processing further comprises resolving embedded references to the embeddable external software elements registered in the software element registry (see Figs 7-9, column 9, line 54 – column 11, line 26).

56. As per claim 29, Jennings teaches wherein the plurality of external software modules comprises software modules supporting external script functions (see Fig 4, column 10, lines 4-6), further comprising:

providing a script function registry for script functions (see column 6, lines 35-40, column 8, lines 53-58, column 10, lines 4-6); and

registering with the script function registry each of the external script functions to be accessible to the electronic content (see column 6, lines 35-40, column 8, lines 53-58).

57. As per claim 30, Jennings teaches wherein the plurality of external software modules comprises software modules supporting at least one of embeddable external visual elements having a visual component for display on a display device and external script functions (see column 5, lines 52-65, column 6, lines 40-42, column 9, lines 18-20), further comprising:

providing a visual element registry and a script function registry for visual elements and script functions, respectively ns (see Fig 3, item 310, column 6, lines 35-40, 45-47, column 8, lines 53-58, column 9, lines 23-61, column 10, lines 4-6, column 11, lines 15-26); and

registering with the visual element registry and the script function registry, respectively, each of the external visual elements and external script functions to be accessible to the electronic content (see column 6, lines 35-40, 45-47, column 8, lines 53-58).

58. Claims 18 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Jennings (US 6,717,593 B1), in view of Peterka et al., (WO 00/30346, hereinafter Peterka), as applied to claims 1 and 17 above, further in view of Peyer (US 6,188,401 B1).

59. As per claim 18, Jennings and Peterka do not teach that the software application is a television program schedule application.

Peyer teaches creating user interfaces for a television schedule application using HTML and JavaScript (see column 1, line 60-67, column 4, lines 44-56, column 6, line 37 – column 7, line 5).

It would have been obvious to one of ordinary skill in the art at the time of the invention that the method of developing user interfaces disclosed in Jennings modified by Peterka could have been applied to television program schedule applications as in Peyer because the method can be applied to a variety of user interfaces and facilitates the creation of user interfaces for new applications (see column 2, line 66 – column 3, line 1; column 8, lines 6-8 of Jennings).

60. As per claim 19, Jennings and Peterka do not teach implementing the method of claim 1 in a television receiver.

Peyer teaches creating user interfaces for a television receiver using HTML and JavaScript (see column 1, line 60-67, column 4, lines 44-56, column 6, line 37 – column 7, line 5).

It would have been obvious to one of ordinary skill in the art at the time of the invention that the method of developing user interfaces disclosed in Jennings modified by Peterka could have been applied to applications for television receivers as in Peyer because the method can be applied to a variety of user interfaces and facilitates the creation of user interfaces for new applications (see column 2, line 66 – column 3, line 1, column 8, lines 6-8 of Jennings).

Conclusion

61. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Sanderson (US 2002/0101448 A1) is cited to teach generating a declarative user interface.
- Bouleau (US 2003/0043192 A1) is cited to teach dynamically modifiable user interface.
- Inanoria (US 2004/0046789 A1) is cited to teach extensible user interface framework and development environment.
- Zaika et al. (US 2004/0056894 A1) is cited to teach a system and method for describing and instantiating extensible user interfaces.

62. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jue Wang
Examiner
Art Unit 2193

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100